

HLA SERVICES OVERVIEW

Aimed At Scalability And Consistency Of Large-Scale Virtual Environments

INTRODUCTION

This paper describes the basic differences between the Distributed Interactive Simulation (DIS) and the High Level Architecture (HLA) and the reasons why DIS has limited applicability in increasingly complex and demanding distributed simulation environments. The overview is written for an audience that has a basic understanding of computer networking and simulation software applications.

HISTORICAL PERSPECTIVE

The DIS standard for distributed simulation was developed to provide a mechanism for tying together loosely coupled training simulations. Like the Simulator Networking (SIMNET) protocol before it, the DIS standard is targeted at loosely coupled training simulations executed on high-speed local area networks (LANs). Loosely coupled training simulations are characterized by imprecise coordination between a few participants (i.e., firing a guided missile at an enemy aircraft at a distance of several hundred meters) and relative insensitivity to message loss or delays. In contrast, simulation of tasks that require precise coordination between several participants on short time scales (e.g., as is required by formation flying (OTA 1995)) are generally considered to be outside the capabilities of the DIS protocols.

In spite of its training focus, DIS has since been applied to test and evaluation activities, analytical simulation experiments, multi-player computer games, and many other uses. (RAND 1996) provides a critical look at DIS and outlines credible uses of the technology. DIS has been very successful in the training domain and, to a lesser extent, in the area of test and evaluation. The application of DIS to large-scale virtual and constructive simulations (e.g., to support theater-level training (Weatherly 1996)), to support simulations of precise interactions (e.g., simulated ball games (Zhou 2002)) and to support massive distributed virtual environments (Macedonia 1994), has been less successful. In many cases, researchers have identified the root causes of failures in these domains, and they have produced technology that addresses these causes.

A decade after the introduction of DIS, HLA has incorporated much of this technology under a new standard for distributed simulation. The set of services called out in the HLA standard are supported by substantially more complicated software technology than is required to implement the DIS protocols. It is this increased complexity that has given rise to simulation middleware such as the HLA Run-time Infrastructure (RTI).

This paper provides an introduction to the time management, declaration management, and Data Distribution Management (DDM) services that are described in the HLA standard. A brief discussion of the technical histories that led to the development of these technologies and their adoption by the HLA community is included as well. It is hoped that this will provide the reader with insight into the contexts in which these services are thought to be useful. The relative sophistication of these services with respect to the standard DIS implementations will highlight the need for RTI software that can encapsulate complex distributed computing services in a reusable package.

DECLARATION AND DATA DISTRIBUTION MANAGEMENT

The original SIMNET, and later DIS protocols employed a broadcast mechanism to disseminate state and event information. Under DIS, the User Datagram Protocol (UDP)/Internet Protocol (IP) was selected as the standard data transport mechanism. One advantage of this approach is that it is easy to implement on a LAN. To send an event or state update, the simulation sets the appropriate broadcast bits in the UDP/IP packet and then places the packet on the network. The network hardware and software ensures that the packet is delivered to all other computers that are connected to that network.

From the simulation developer's point of view, the broadcast-based DIS protocol limits scalability in the following two ways.

First, each simulation must process every packet that is placed on the network. For large-scale exercises, the resources required for a computer to handle thousands of data packets per second can be enormous (Macedonia 1995). For example, in a 1000-entity exercise conducted in 1990 with SIMNET, the limiting factor was not network bandwidth, but the local host processor performance. Improvements in Central Processing Unit (CPU) and memory performance in the last decade have, in part, served to offset this. Nonetheless, the 1994 Synthetic Theater of War-Europe (STOW-E) exercise, conducted with about 1800 entities, exhibited a similar limitation.

Second, the amount of network bandwidth required to conduct a large-scale exercise is also an issue with the DIS protocol. The introduction of Gigabit Ethernet and other high-speed LAN technology makes considerable strides in addressing this concern for a small-scale exercise. However, current Wide Area Network (WAN) technology is not on par with the performance of LAN technology. This presents an obstacle to widely distributed, large-scale DIS exercises. For example, a 100,000-entity simulation exercise that uses DIS would require 375 Megabits per second (Mbps) of network bandwidth to be available for each computer involved in the exercise (Macedonia 1995). While a Gigabit LAN might be able to support such an exercise (although the computers attached to the network probably could not), it is unfeasible with currently available WAN technology.

Interest management technology is one solution to the data explosion problem that is endemic to DIS and DIS-like simulation protocols. Interest management technology limits the delivery of data to just those systems that can make effective use of the data. Numerous interest management schemes have been proposed over the course of the last decade. The HLA has adopted two distinct mechanisms to support interest management. These are a publish/subscribe distributed event notification system (declaration management services) and space-based interest management (DDM services).

The HLA incorporates a publish/subscribe-based distributed event system. The Federation Object Model (FOM) describes the types of data available within a federation. Data types can be objects or interactions. An object has an object class name and a set of attributes. Interactions have an interaction class name and a set of parameters. The objects and interactions in an object model are related by specialization/generalization (also known as “inheritance” and “is-a”) relationships. An HLA federate expresses interest in events of a particular type by subscribing to the object class and attribute set or interaction class that describes the event type. Similarly, a federate expresses its ability to provide certain types of events by declaring its ability to publish the appropriate object classes and attributes and interaction classes. The HLA RTI can take advantage of publication and subscription information to ensure that events of a particular type are transported only to those federates that have expressed interest in those event types.

For example, suppose a ground-attack aircraft and a Surface-to-Air Missile (SAM) battery are participating in simulated battle involving sea-based, ground-based, and aerial vehicles. Physical entities might be generically described by a “BaseEntity” class that has the attributes “Position” and “Velocity.” Land-based entities might be described by the class “GroundVehicle,” which is a specialization of “BaseEntity,” whose attribute set includes everything in “BaseEntity” plus a field “CamouflagePattern.” Similarly, a “SeagoingVessel” class that specializes “BaseEntity” and adds the attribute “Orientation” describes sea-based entities. Finally, aerial vehicles could be described by an “AirVehicle” class that also specializes “BaseEntity” and adds an “Orientation” attribute.

The ground-attack aircraft might subscribe to the “GroundVehicle” class and all its attributes, but not to the “SeagoingVessel” or “AirVehicle” classes. The HLA RTI would deliver only state information published as “GroundVehicle” data to the ground-attack simulator, thereby restricting the network and computational load at that federate to the ground targets of interest. Similarly, the SAM battery would likely subscribe to just the “AirVehicle” class, thereby restricting its data processing to potential aerial targets.

The origin of the HLA’s distributed event services can be seen in past and current distributed event systems. For example, the Event-Constraint-Object (ECO) distributed event system (Haahr 2000) demonstrates how a publish/subscribe mechanism can be used to significantly reduce network traffic in large-scale distributed

computing systems. The ECO software uses multicast groups and pattern-matching to prevent the delivery of unwanted data. The ECO system was embedded in a distributed monitoring system and was seen to reduce network traffic by as much as 80 percent with respect to a broadcast-based implementation.

The HLA incorporates space-based DDM via its DDM services. The HLA DDM services have their origins in the space-based interest management schemes demonstrated as part of the Naval Postgraduate School Network (NPSNET) (Macedonia 1995, Macedonia 1994), early STOW exercises, research in distributed virtual environments such as Distributed Interactive Virtual Environment (DIVE) (Carlsson 1993), and numerous other systems (for a survey, see (Morse 1996)). The scheme is conceptually simple. When simulating a collection of physical entities, any particular entity can only react to and act upon events that occur within a certain distance of the entity. For example, a tank is likely to be interested in an area no more than 10 kilometers in diameter. A dismounted infantryman may have an Area of Interest (AOI) no larger than a few hundred meters.

In the HLA, areas of interest are described by routing spaces. A routing space is a rectangular region in space. A federate can subscribe to events of a particular type that occur within particular regions. A typical use of routing spaces is to divide a simulated battlefield into cells. As entities move through space, they change their subscriptions to match the cell(s) that they presently occupy.

Consider the ground-attack aircraft from the previous example. Suppose the aircraft can detect tanks that are in any of the eight cells surrounding the aircraft. As the aircraft moves through space, it un-subscribes to ground-based state updates in the cells that are no longer visible and subscribes to updates in the set of visible cells. Figure 1 depicts the motion of the aircraft through space and its AOI at two points in time. At time t_1 , the aircraft will receive "GroundVehicle" events generated by Tank A. At time t_2 , the aircraft will receive "GroundVehicle" events generated by Tank B. It will never receive updates from Tank A and Tank B simultaneously.

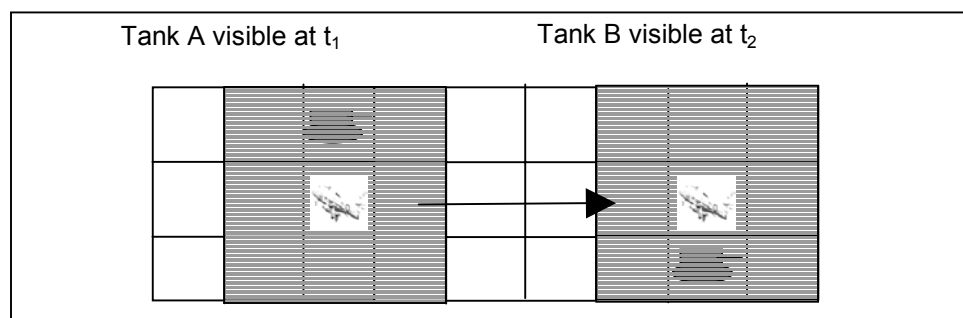


Figure 1. The aircraft changes its subscription region as it moves through space.

For the interested reader, (Boukerche 2000) provides an excellent overview of the HLA DDM system.

TIME MANAGEMENT AND MESSAGE ORDERING SERVICES

The DIS standard calls for synchronized real-time clocks to provide a common time reference for the federated simulations. In general, a 100-millisecond latency is sufficiently small to ensure that human participants in a simulation exercise are not able to perceive latency-induced effects. However, this number was produced when SIMNET, and later DIS, was being used on LANs, where out-of-order message delivery was uncommon and message loss rates were relatively small (on the order of 2 percent).

In the military arena, the first-distributed time-managed training simulation was based on the Aggregate Level Simulation Protocol (ALSP) (Weatherly 1996). ALSP had its origins in the unsuccessful use of a SIMNET-like protocol to distribute the Ground Warfare Simulator (GRWSIM) over a WAN for use in the Allied Command Europe (ACE)-89 computerized military exercise. The distributed GRWSIM software comprised the major computational component of the exercise. Unfortunately, unreliability of the central portion of the GRWSIM game, and inconsistencies in the perceived state of the simulated world, had profound effects on the state of combat. Players in the simulation exercise were frequently faced with tactical and strategic challenges that were the result of simulation artifacts (i.e., the result of physically distributing the simulation software) rather than enemy action.

A subsequent analysis of the ACE-89 event revealed several requirements, beyond those of SIMNET, that were incorporated in the ALSP. Among these was a need for rigorous simulation time management.

The time management schemes adopted by the ALSP, and subsequently by the HLA, are based on research in the field of parallel discrete event simulation (PDES). The goal of time management in PDES systems is to ensure that events at a process occur in time-stamp order. Several successful time management schemes had been developed by 1989, when the ALSP was first conceived, and had been used by the high-performance computing community for several years. ALSP adopted a conservative scheme, which it used for several years, before moving on to more complicated (and more capable) schemes that were subsequently incorporated into the HLA standard.

The HLA incorporates three different time management schemes. The first of these is a receive order scheme. With a receive order scheme, messages are delivered to the federate as they arrive. The second scheme ensures causally consistent message ordering. A sequence of events is said to be causally ordered if it is arranged in such a way that always causes preceded effects. Events that are causally unrelated can appear in any order (relative to each other).

The effect of causally consistent message order can be best demonstrated with an example. Suppose there is a hill that separates two artillery batteries and that a counter-battery radar is assigned to one of the batteries. Consider the following

sequence of events. The enemy battery fires on the counter-battery radar. The radar detects the counter-battery fire and issues a position report to its associated battery. Moments later, the counter-battery radar is destroyed. Finally, the friendly battery uses the (deceased) radar report to launch a counterattack.

In a distributed simulation, this could correspond to the following sequence of messages. The enemy counter battery issues a fire interaction. The fire interaction is received by a munitions model that computes the impact time of the round. The counter-battery radar receives the fire interaction and generates a report for its associated radar. The counter battery sends the report. Finally, the counter battery detects a munitions detonation generated by the munitions simulator (the enemy rounds have arrived) and the radar is destroyed. This event sequence is shown in figure 2. In this scenario, the fire interaction causally precedes the munitions detonation. A causally consistent message delivery system will ensure that all federates see the fire interaction and munitions detonation message in the correct causal order.

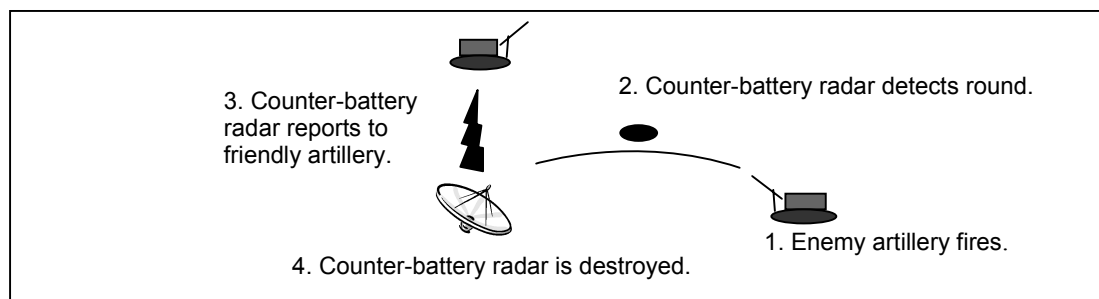


Figure 2: Simulation of a counter-battery scenario

Suppose the enemy fire interaction is delayed in the network, resulting in the munitions detonation event arriving first at the counter-battery radar federate. Note that a simulation system that uses a transport mechanism which guarantees First-In, First Out (FIFO) delivery (e.g., TCP/IP) is still subject to such an occurrence because the fire interaction and munitions detonation messages are generated by two distinct processes. If this were to occur, the outcome of the engagement would be radically different. In this case, the counter-battery radar is destroyed before it has an opportunity to detect the incoming rounds and request a counterstrike! The enemy battery survives because of a simulation artifact, whereas it would have been killed if causally consistent message delivery had been used.

The third HLA time management scheme ensures strict time-ordered delivery of messages. This is a stronger ordering condition than causally consistent ordering. Time-ordered delivery ensures that a federate sees simulation events in strict time-stamp order. For example, a causally consistent event ordering in our previous counter-battery example would allow an observer to see the friendly battery launch its counterattack at any point following the receipt of the radar report. Two runs of the scenario could produce two slightly different results. In the first run, the radar might be destroyed before the counterstrike is launched. In a second run, the radar might be destroyed after the counterstrike is launched. This is because the munitions detonation

event and the counterstrike are causally independent. However, if a time-stamp ordering of events is used, any number of executions of the example scenario will always produce the same results.

In addition to these basic time management schemes, the HLA provides support for speculative message release and message retraction, mixed real-time and faster than real-time simulation, and several other advanced time management concepts. See (Fujimoto 1996) for a detailed overview.

RTI INTEROPERABILITY

The HLA standard describes a set of distributed computing services and an Application Programmer's Interface that can be used to access those services. This is aimed at source-code level compatibility amongst different RTI vendors. The HLA standard does not specify "under the hood" implementation details. This effectively prohibits run-time interoperability between RTI software provided by different vendors.

The ability to specify a standard RTI implementation is complicated by the need for RTI software not only to exchange messages, but also to encapsulate a distributed system state that is needed to implement many of its more sophisticated services. Some concrete examples of issues faced in trying to construct standards for implementing these services are listed below.

The implementation of time ordered message delivery requires the use of so-called "global virtual time" distributed algorithms. There are several such algorithms available for use by RTI software designers, none of which has risen to a dominant position.

Numerous implementations of declaration management services have appeared over the years. A key feature of a declaration management service is how interest information (i.e., publications and subscriptions) is distributed. Some options include a centralized interest database, receiver filtering, assignment of interest categories to multicast groups, and broadcast-based dissemination of interest information in conjunction with sender filtering.

Lastly, we can mention DDM services. A common solution is to assign logical areas of interest to multicast groups. The assignment of multicast addresses to areas of interest is implementation dependent. Also note that this might conflict with the use of multicast groups for declaration management by a different RTI.

A discussion of RTI interoperability heads naturally toward HLA/HLA gateways. The use of HLA/HLA gateways has been proposed as a solution for achieving FOM agility, enabling information hiding, and to allow otherwise non-interoperable RTI software to exchange data. The RTI Interoperability Study Group provides an in-depth report on the potential capabilities and pitfalls of employing HLA/HLA gateways.

Most commonly, the use of gateways reduces a federation's capability in order to mediate the need to modify legacy systems. At the very least, gateways introduce additional end-to-end latencies due to processing overhead. In large-scale systems, federation gateways can act as performance bottlenecks. This can occur, for instance, when a gateway is used to bridge large federations (Griffin 1997), as shown in figure 3.

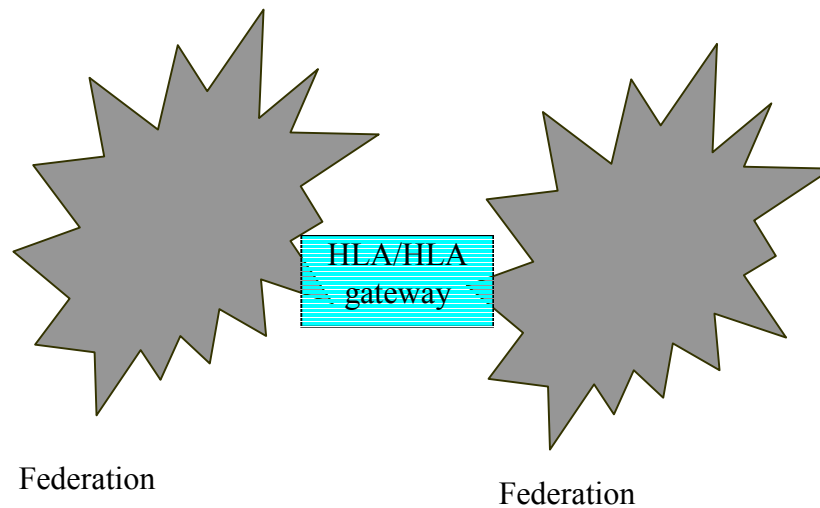


Figure 3: Two federations connected by an HLA/HLA gateway.

Capability losses are not only performance related. The use of gateways can prevent a federation designer from employing some HLA services. For example, the HLA time management services can only be employed across an HLA/HLA gateway in special cases using federation specific software. DDM and declaration management services can also be restricted or their effectiveness significantly diminished when an HLA/HLA gateway is encountered (Dingel 2001).

RTI-SPECIFIC ISSUES

At present, there are four Defense Modeling and Simulation Office (DMSO) certified and two Institute of Electrical and Electronic Engineers (IEEE) certified RTI implementations (<http://www.dmsomil/public/transition/hla/rti/statusboard>). The DMSO site lists 17 RTIs; however, some are only different version updates of the same RTI.

There are several reasons why there are so many RTIs. For example, there are two standards. The DMSO standard is the original, and there are several versions, from RTI Next Generation (NG) 1.3v2 to 1.3v6. The range represents the evolution of the standard since its inception. There is also the IEEE standard, which has two versions currently certified. The DMSO and the IEEE standards are NOT interchangeable. The DMSO standard addresses the traditional and evolving needs of the military community and the IEEE standard is intended to expand into other non-military domains such as the automobile, aircraft, and entertainment industry.

There are multiple platforms that can host an RTI and multiple Application Programming Interfaces (APIs) for development. Table 1 (see the appendix) highlights key features of some commercially available RTI technology.

The multiple platforms and multiple APIs provide flexibility for federation developers, but also create interoperability issues. This is analogous to source code/compiler issues in the mainstream software development domain. For example, there are many different C++ compilers for many different operating systems (OS), but there are often interoperability or porting issues as source code moves from one compiler on one OS to a different compiler on a different OS. Just as traditional source code must be carefully adapted when moving across platforms or compilers, federations must be designed to take into account the standard to which a particular RTI adheres and what services will be used. API and Platform choice, as well as network issues, also impact federation designs and may have a significant impact on functionality and/or performance.

It is important to note that FOMs written for one RTI may not work with another, and may require either relatively small changes or complete rewriting, depending on what is expected from the federation in terms of functionality and performance.

RTI performance is only one part of the overall HLA performance picture in a similar way that compiler performance in the traditional software development domain is a combination of good source code design and compiler choice. This is not to say that there are no performance differences between RTIs that may affect federation performance. In fact, we strongly encourage performance testing of RTIs, but with the understanding that this is only one of several important performance variables.

PERFORMANCE METRICS FOR RTI SOFTWARE

The performance of a distributed simulation system can be characterized in several ways. Some measures of performance include bandwidth usage (i.e., message volume), computational overhead at each node, memory overhead at each node, and rate of advance of the simulation clock. Overall federation level performance is strongly influenced by federation design; capabilities of the host computers; the environment (LAN or WAN); the scale, i.e., the number of federates; and the scope of the federates, i.e., what the federates do, their attributes, etc.

In order to develop performance benchmarks for RTI software, it is necessary to identify the scale and scope of the federations at which the benchmarks are to be targeted. The HLA provides federation designers with a relatively large (with respect to DIS) design space. The importance of careful federation design to federation performance cannot be overemphasized. A federation design should be targeted at a well-understood federation scale and scope. Consequently, RTI benchmarks must be similarly targeted if they are to provide meaningful guidance to federation designers and policymakers.

A generic description of candidate federates should include, at least, the following features:

1. RTI services of interest (e.g., time management and DDM).
2. The number of federates that will participate in an exercise.
3. The number of simulated entities, their distribution amongst participating federations, and descriptions of key interactions.

A reference FOM can be used as an approximation of federation designs that will be derived from the reference FOM. The reference FOM will provide information about required RTI services (e.g., the Real-time Platform Reference [RPR]-FOM restricts itself to federation management, object management, and declaration management services). The metadata associated with a FOM, in conjunction with its object model proper, should serve as a record of the intended scale and scope of federations derived from the reference FOM.

For example, the RPR-FOM describes itself as having a scale and scope that exactly matches the scale and scope of the DIS protocol. This suggests that federations whose FOM is derived from the RPR-FOM are interested in performance metrics that are relative to DIS-like federations. In particular, end-to-end latency and bandwidth usage are considered key measurements of RTI performance in this context.

A consumer of benchmark data must be wary of extrapolating benchmark results generated for a particular FOM. For example, a change from best effort to reliable, causal, or time-stamp ordered delivery of commonly used attributes (e.g., entity position) or parameters can have significant effects on federation performance. This effect can be positive if, say, some particular global consistency or event rate goal is of paramount importance (e.g., as might be required for campaign or theater level war games). On the other hand, if small end-to-end latencies are critical (e.g., in a closed-loop Hardware-in-the-Loop [HWIL] test), the effect of such a change might be perceived as negative.

To further illustrate the importance of considering context while interpreting benchmark results, consider the use of data distribution services. The benefit of applying DDM is a function of message density (volume of network traffic over time) and available bandwidth. For small-scale federations, a degradation of end-to-end latency performance may not be matched by significant bandwidth savings. For mid-scale federations, a nearly even tradeoff might be observed. For very large federations, significant bandwidth savings might result in both decreased latencies and reduced packet loss rates. The location of cutoff points (e.g., from small to mid-scale to large) is related to the network technology being utilized, available processing power, and internal federate behavior (e.g., in a space-based DDM system, what is the frequency of AOI boundary crossings?).

The boundaries of the experimental frame we choose will determine the applicability and extensibility of RTI performance characteristics to other contexts. A detailed experimental design for this effort is underway and will be produced in an accompanying document.

CONCLUSIONS

The DIS protocol, like its SIMNET predecessor, is targeted at loosely coupled training simulations executed on high-speed LANs. Throughout the 1990s, researchers working on large-scale distributed virtual environments discovered that the relatively simple approach taken by the early DIS and DIS-like protocols did not scale to large groups, nor did it effectively support small groups participating in tasks requiring precise coordination.

In the military arena, the ALSP emerged in response to demonstrated shortcomings of the DIS protocol. Meanwhile, researchers in industry and academia sought to develop technology that would enable a new generation of virtual reality applications. In the late 1990s, the HLA incorporated much of this technology under a new standard for distributed simulation.

Effective use of the HLA requires that federation designers be aware of the services that the new standard provides and the context in which they can be applied. To treat the HLA as a replacement for DIS is to discard those features of the HLA that justify a technology migration in the first place. However, misuse of the HLA services can result in systems that do not perform adequately, or which exhibit undesirable behavior. Careful federation design by knowledgeable federation designers is a prerequisite for any successful distributed simulation project.

REFERENCES

(Boukerche 2000) A. Boukerche and A. Roy, "In Search of Data Distribution Management in Large Scale Distributed Simulations," Proc. of the 2000 Summer Computer Simulation Conference, Vancouver, Canada, pp. 21-28, 2000. Available at <http://www.cs.bham.ac.uk/research/pdesmas/>

(Carlsson 1993) Carlsson and Hagsand, "DIVE—A Multi User Virtual Reality System," IEEE VRAIS, Sept, 1993. Also see the DIVE website at <http://www.sics.se/dive/>

(Dingel 2001) Juergen Dingel, David Garlan, Craig A. Damon, "A Feasibility Study of the HLA Bridge," CMU-CS-01-103, Computer Science Department, School of Computer Science, Carnegie Mellon University, March 2001. Available on the web at <http://reports-archive.adm.cs.cmu.edu/anon/2001/abstracts/01-103.html>

(Fujimoto 1996) R. M. Fujimoto and R. M. Weatherly, "HLA Time Management and DIS," 14th Workshop on Standards for the Interoperability of Distributed Simulation, March 1996. Available at <http://www.cc.gatech.edu/computing/pads/tech-highperf-ddm.html>

(Griffin 1997) Sean P. Griffin, Ernest H. Page, Zachary Furness, Mary C. Fischer, "Providing Uninterrupted Training to the Joint Training Confederation (JTC) Audience During Transition to the High Level Architecture (HLA)," Simulation Interoperability Workshop, Fall 1997. Available on the web at http://alsp.ie.org/alsp/biblio/SIW_Fall97/ADAPTOR_SIW97.html

(Haahr 2000) M. Haahr, R. Meier, P. Nixon, V. Cahill, and E. Jul, "Filtering and Scalability in the ECO Distributed Event Model," in Proceedings of the 5th International Symposium on Software Engineering for Parallel and Distributed Systems, Limerick, Ireland: IEEE Computer Society, 2000, pp. 83-95. Available at www.dsg.cs.tcd.ie/~meierr/

(Macedonia 1994) Macedonia, Michael R., Zyda, Michael J., Pratt, David R., Barham, Paul T. and Zeswitz, Steven "NPSNET: A Network Software Architecture for Large Scale Virtual Environments," Presence, Vol. 3, No. 4, Fall 1994, pp. 265-287. Available from <http://www.npsnet.org/~npsnet>

(Macedonia 1995) Macedonia, Michael R., Zyda, Michael J., Pratt, David R., Brutzman, Donald P. and Barham, Paul T. "Exploiting Reality with Multicast Groups," IEEE Computer Graphics & Applications (revised from appearance in the VRAIS '95 Proceedings), September 1995, pp. 38-45. Available from <http://www.npsnet.org/~npsnet>

(Morse 1996) Katherine L. Morse. "Interest management in large-scale distributed simulations." Technical Report ICS-TR-96-27, 1996. Available at <http://www.cs.nott.ac.uk/~mhl/archive/archive.html>

(OTA 1995) U.S. Congress, Office of Technology Assessment, *Distributed Interactive Simulation of Combat*, OTA-BP-ISS-151 (Washington, D.C.: U.S. Government Printing Office, Sept. 1995). Available at http://www.wws.princeton.edu/~ota/disk1/1995/9512_n.html

(RAND 1996) "Credible Uses of the Distributed Interactive Simulation (DIS) System," James A. Dewar, Steven C. Bankes, James S. Hodges, Thomas Lucas, Desmond K. Saunders-Newton, Patrick Vye, MR-607-A, 1996. Available on the web at <http://www.rand.org/publications/MR/MR607.pdf>

RTI Interoperability Study Group Final Report: SISO-REF-003-1999 (99F-SIW-001). Available on the web at http://www.sisostds.org/doclib/doclib.cfm?SISO_CID_61

(Weatherly 1996) R.M. Weatherly, A.L. Wilson, B.S. Canova, E.H. Page, A.A. Zabek, M.C. Fischer, "Advanced Distributed Simulation Through the Aggregate Level Simulation Protocol," Proceedings of the 29th Hawaii International Conference on System Sciences, Volume 1, pp. 407-415, Wailea, Hawaii, 3-6 January 1996. Available on the web at <http://alsp.ie.org/alsp>

(Zhou 2002) Suiping Zhou, Wentong Cai Nanyang, Stephen J. Turner, Francis B. S. Lee, "Critical causality in distributed virtual environments," Proceedings of the sixteenth workshop on parallel and distributed simulation, pp. 53-59, 2002.

APPENDIX

RTI Name	Vendor	HLA Spec.	Language	Platform	Compiler	Price	Notes
pRTI	Pitch	1516, 1.3	Java2, C++	Java2, Windows NT/2000	Java2, MS Visual C++	~ \$1000 per federate	The 1516 version has been certified by DMSO. Website is at http://www.pitch.se/
ERTI	Mitsubishi Space Software Co., Ltd. (MSS)	1.3	C++	SunOS 8, Windows NT/2000, Linux (kernel 2.4.7)	Sun Visual Workshop C++ Compiler, MS Visual C++, GNU C++ Compiler 2.95.3		
MAK High Performance RTI	MAK Technologies	1.3	C++	Windows NT/2000/XP, Linux (Red Hat 7.2), Solaris, IRIX	MS Visual C++, GNU C++ Compiler 3.0.2	~ \$1250 per federate	The MAK RTI has passed DMSO verification testing based on the 1.3 interface specification. The MAK website is at http://www.mak.com/
RTI-NG Pro	VTC	1.3	C++	Solaris 2.8, Windows 2000, SGI IRIX OS	C++ Forte 6.0 Update 1 for Solaris, MS Visual C++, SGI CC compiler for IRIX		Was previously the DMSO RTI-NG 1.3v6. Support was included for Windows 2000, Solaris, Linux, and the SGI IRIX operating systems. Company website is at http://www.virtc.com/

Table 1. Commercial RTI Information Summary